Network Layer

Transport / Network Interface

Transport layer requests service from network layer

Transport Layer	Prepare Network service requirements Transport segment for transfer across network Application data + transport header Local node address Remote node address
Network Layer	Receives Transport layer PDU — data + transport header Interface Control Information (ICI) Service requirements Local + remote node addresses

Connection Types

Connection

State machine associated with data exchange

Connection-oriented

Set-up end-to-end network channel before data any exchange

Monitor channel state during data exchange

Multiple transactions associated with connection state

Close channel after data exchange

Example — **phone call**

Enter number → answer call → extended conversation → disconnect

Connectionless

Data transmitted with no prior channel set-up

No channel state defined by nodes

Each message independent

Example — **email message**

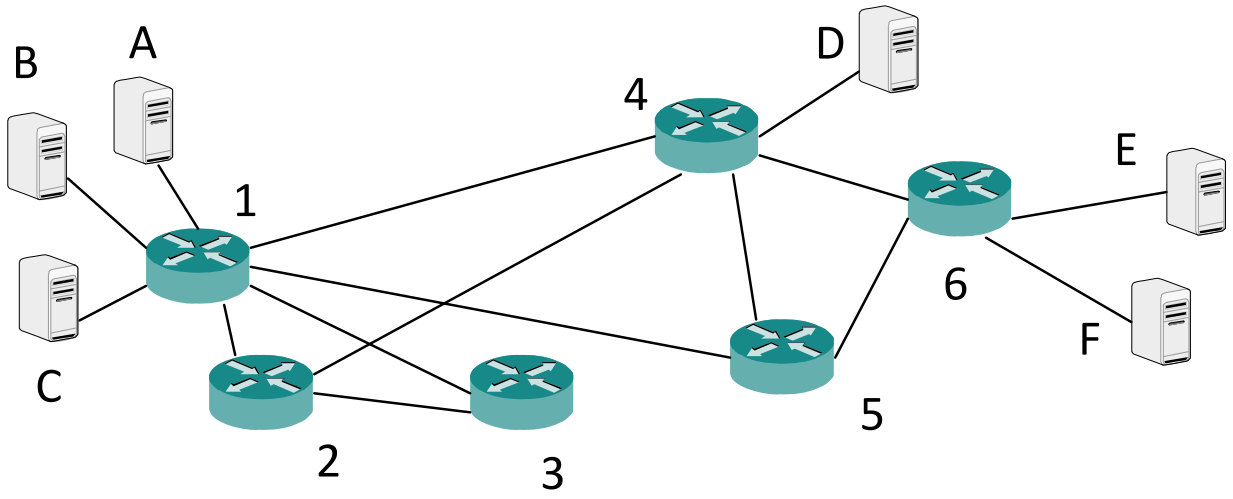
Send email → hope message arrives → hope message is found / read

Datagram Service

Network of routers and links

Packet switching

Connectionless



Every datagram

Has source and destination address in header

- Data Link header or Network header

Routed individually through network

Datagrams may follow separate routes

Best effort delivery — unreliable service at layer 3

Example

src = B dest = F	data

B → 1 → 4 → 6 → F

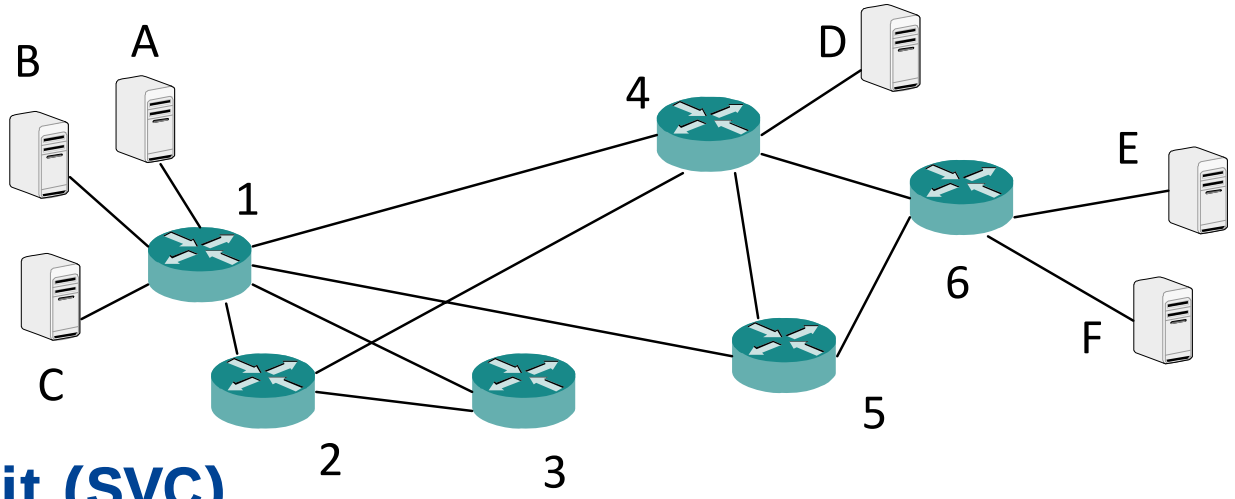
B → 1 → 5 → 6 → F

Switched Virtual Circuit (SVC)

Network of switches and links

Circuit switching or packet switching

Connection-oriented



Switched Virtual Circuit (SVC)

Set-up / close messages carry source and destination addresses

Example

Set-up VC–1: B → 1 → 4 → 6 → F

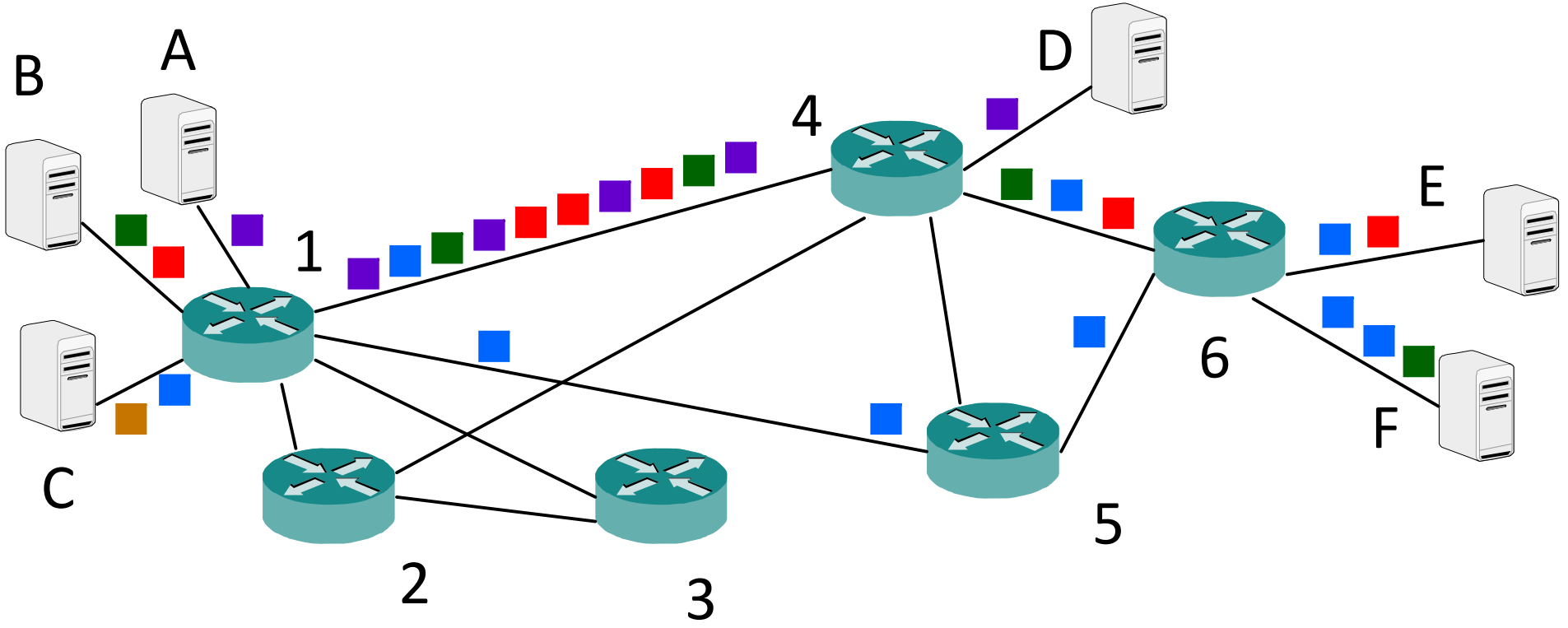
Packet routing by VC ID in header (layer 2 or layer 3)

Every packet follows same VC route

Example

VC – 1	data

Switching Example



A to D — circuit mode (deterministic SVC)

B to E — packet mode (statistical SVC)

B to F — packet mode (statistical SVC)

C to F — packet mode (datagram service)

Virtual Circuit Labeling

Routing

VC number assigned at each switch

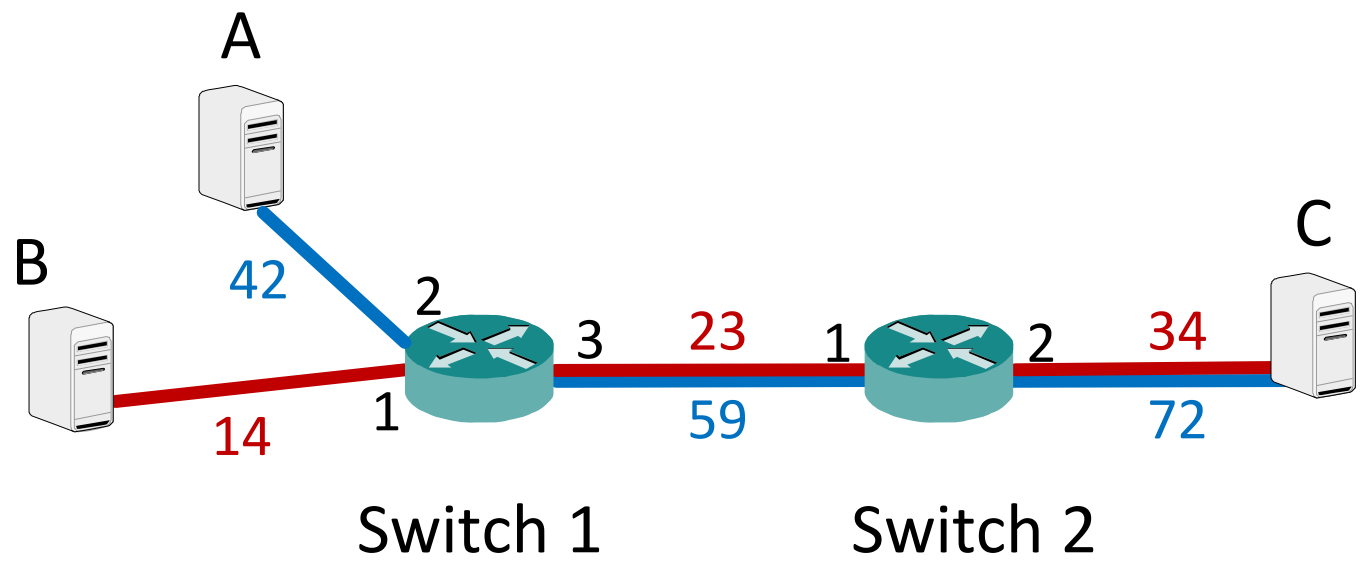
- Source-to-destination label on one hop
- Can change at each switch

VC in packet header identifies path

Example

X.25 network packet

Format	VC	Type	Data



Virtual Circuit Routing Table

Routing pair

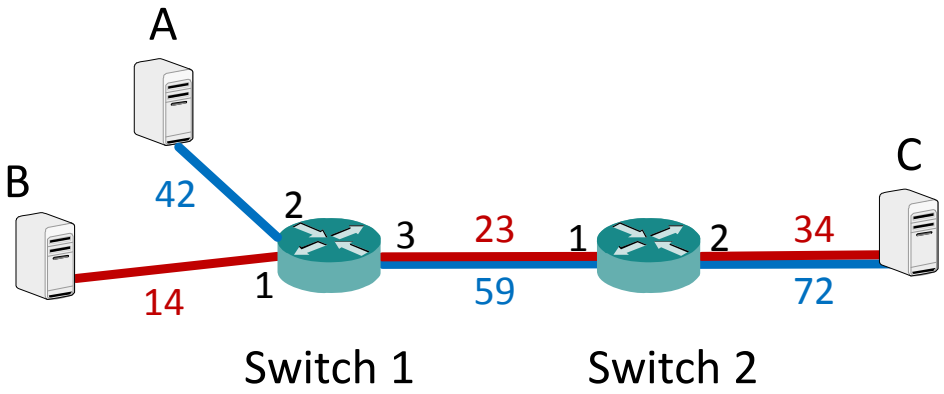
Interface number

VC

Routing table for VC

VC set-up = write table entries

Map input pair → output pair



Switch 1 Routing Table			
Input		Output	
Interface	VC	Interface	VC
1	14	3	23
2	42	3	59
3	23	1	14
3	59	2	42

Switch 2 Routing Table			
Input		Output	
Interface	VC	Interface	VC
1	23	2	34
1	59	2	72
2	34	1	23
2	72	1	59

Datagram Forwarding

Routing

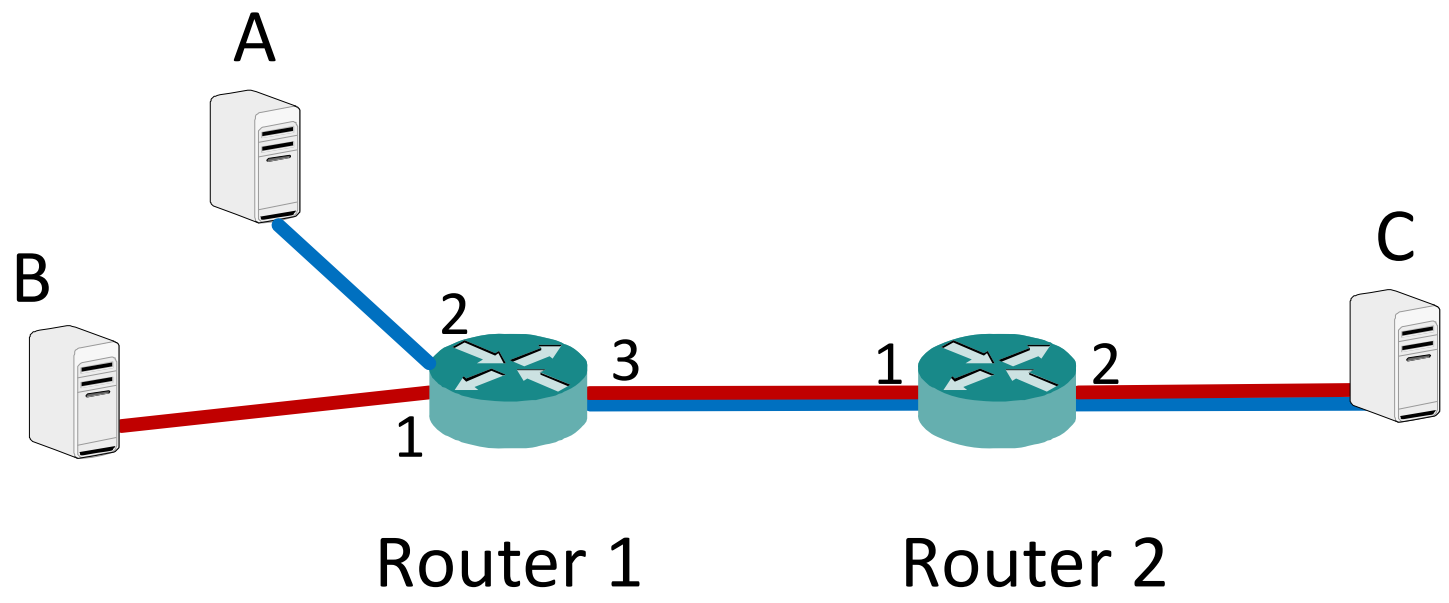
Source address + destination address in IP header

Destination pair identifies next hop

Example

IP

Src	Dest	Other Fields	Data

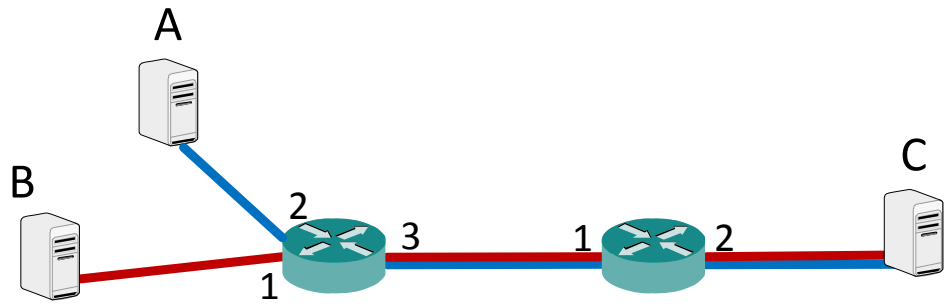


Datagram Forwarding

Routing table for datagram

Routing algorithm → table entries

Map destination address → output interface



Router 1 Table	
Destination Address Range	**Interface**
A	2
B	1
Other	3

Router 2 Table	
Destination Address Range	**Interface**
C	2
Other	1

Routing algorithm

1 tells 2 "I have A + B"

2 tells 1 "I have C"

Connection-Oriented / Connectionless Networks

Connection-oriented network services

ESS#7	64 kbps pubic switched telephone network (PSTN)
X.25	64 kbps pubic switched data network (PSDN)
Frame relay	2.048 – 45 Mbps replacement for X.25
ATM	64 kbps – 625 Mbps replacement for ESS#7 + Frame Relay
GSM / UMTS	Cellular voice network
GPRS / EDGE	384 kbps – 1.9 Mbps cellular data network

Connectionless network service

Internet Protocol (IP)

Internet Protocol version 4 (IPv4)

IP datagram format

4 bits	4 bits	8 bits	16 bits
Version	Hlen	Service Type	Total Length (header + data in bytes)
Identification		Flags	Fragment Offset (13 bits)
Time to Live		Protocol	Header Checksum
Source IP Address			
Destination IP Address			
Options			
Data			

Hlen	Header length in 32-bit fields
Service type	Differentiated Services Code Point (DSCP) Explicit Congestion Notification (ECN)
Identification	Provides unique datagram ID
Flags	**0** | **DF** (Don't Fragment) | **MF** (More Fragments — all frags but last)
Fragment Offset	Offset in 8-byte units from start of original datagram (fragmented)
Time To Live (TTL)	Routers perform: **`{if (--TTL == 0) delete datagram}`**
Protocol	Protocol of data carried by datagram

Internet Protocol version 4 (IPv4)

Fragmentation & Reassembly

Maximum Transmission Unit (MTU)

Largest datagram accepted by Layer 2 links

Fragmentation

Divide datagram longer than MTU into < 2^{13} = 8192 datagrams

If DF = 1 datagram discarded

Reassembly

Original datagram reassembled at final destination

Example

MTU = 1500 bytes = 20 (header) + 1480 (data)

Total Length = 4000 bytes = header + 3980

3980 – 2 * 1480 = 1020

3 fragments with original header except

1 — Length = 1500 bytes MF = 1 offset = 0

2 — Length = 1500 bytes MF = 1 offset = 1480 / 8 = 185

3 — Length = 1040 bytes MF = 0 offset = 2*1480 / 8 = 370

Internet Protocol version 4 (IPv4)

IPv4 Addressing

Internet address

32-bit address

4 octets (8-bit decimal numbers) separated by dots

Example

www.hadassah.ac.il ↔ 212.179.79.228

Network / Host

Host address = 212.179.79.228

Host address = Network.Host

Network number = 212.179.79.0

Host number = 228

Network / Subnet / Host

Divide hosts into sub-networks

$$\text{Host number} = \begin{cases} 0, \ldots, A-1 & \text{(subnet 0)} \\ A, \ldots, 2A-1 & \text{(subnet 1)} \\ 2A, \ldots, 3A-1 & \text{(subnet 2)} \\ \quad \ldots & \end{cases}$$

Example

Subnet 0 = 212.179.79.1 – 212.179.79.127

Host address range = 212.179.79.X

X = 0xxxxxxx (binary)

Subnet 1 = 212.179.79.128 – 212.179.79.254

Host address range = 212.179.79.X

X = 1xxxxxxx (binary)

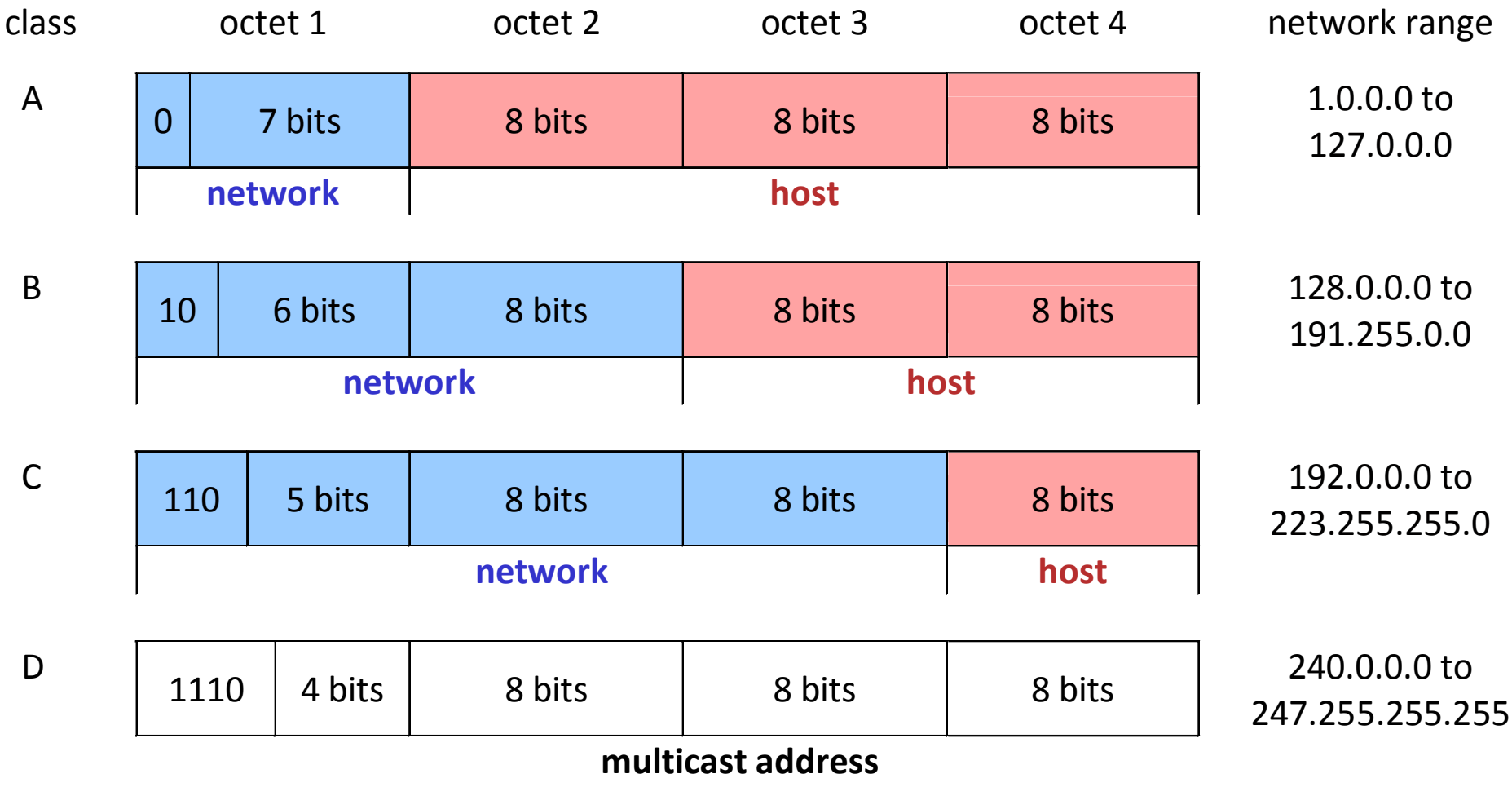
Internet Protocol version 4 (IPv4)

Address classes

3 main address classes

A — small number of large networks (up to 2^{24} = 16 Mhosts)

C — large number of small networks (up to 2^{8} = 256 hosts)



Internet Protocol version 4 (IPv4)

Classless Inter-Domain Routing (CIDR)

Address format

Octet1.Octet2.Octet3.Octet4/bits_in_network_number

bits_in_network_number = 1, 2, 3, … , 31 (unrestricted)

Examples

Class A address — **10.0.1.5/8**

8-bit network number = **10.0.0.0**

Class C address — **192.168.0.37/24**

24-bit network number = **192.168.0.0**

General node address — **192.168.0.33/27**

192	168	0	33	
11000000	10101000	00000000	001	00001
27-bit network address				host

Network address = **192.168.0.32** Host number = **0 – 31**

Host addresses = **192.168.0.32 – 192.168.0.63**

Internet Protocol version 4 (IPv4)

Subnet mask

Forming subnet mask

1 in all bits of network number

0 in all bits of host number

Encodes same information as number of bits in network number

Example

General node address — **192.168.0.33/27**

27-bit network number

Mask

11111111.11111111.11111111.11100000

255.255.255.224

Using mask

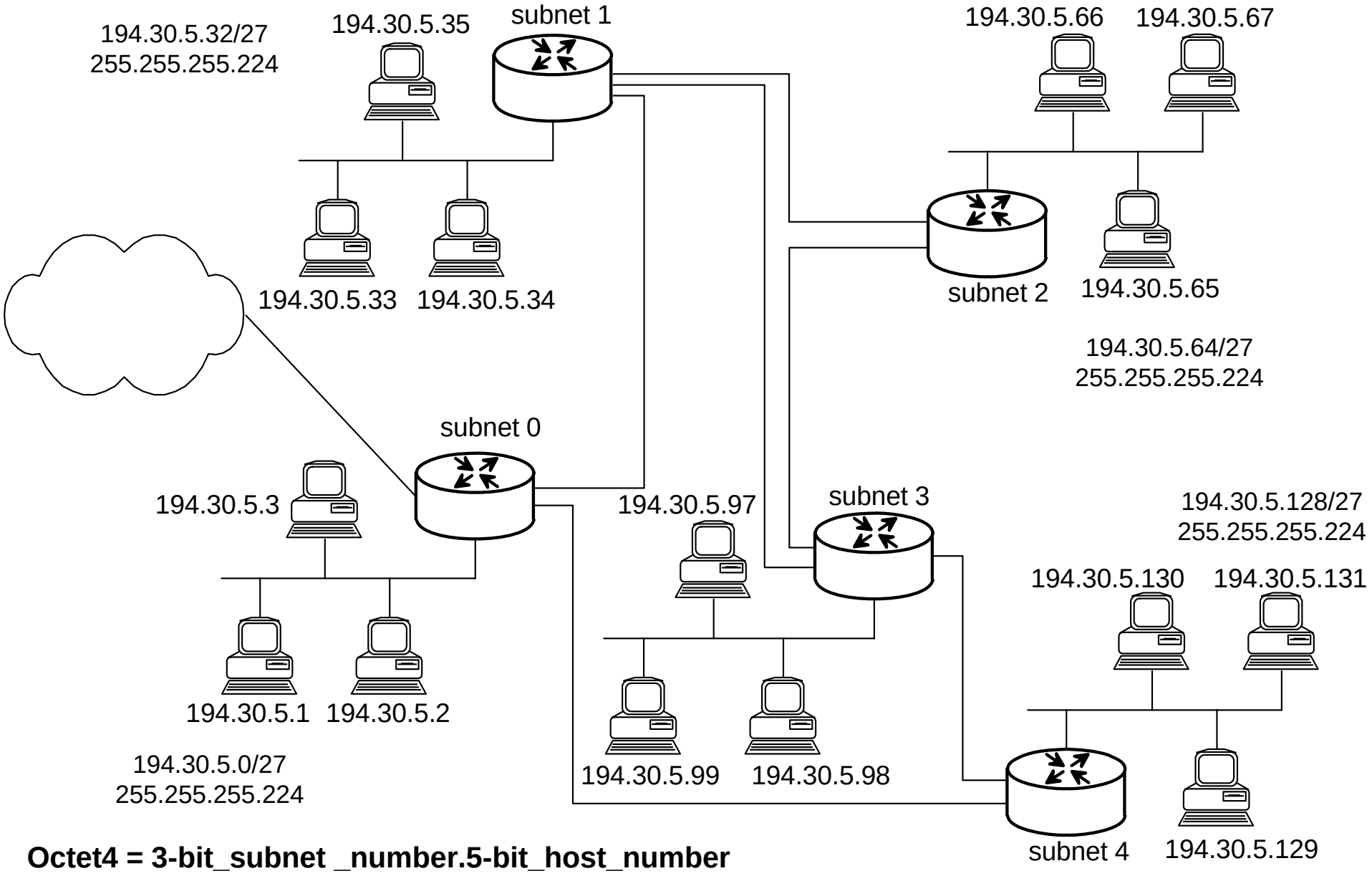
Mask AND IP address = network number

Example

255.255.255.224 AND 192.168.0.33 = 192.168.0.32

Internet Protocol version 4 (IPv4)

Subnet example



Octet4 = 3-bit_subnet _number.5-bit_host_number

Internet Protocol version 4 (IPv4)

Reserved addresses

CIDR address block	Description
0.0.0.0/8	Current network (source address)
10.0.0.0/8	Private network
127.0.0.0/8	Loopback (destination = this node)
172.16.0.0/12	Private network
192.168.0.0/16	Private network
224.0.0.0/4	Multicast (Class D)
240.0.0.0/4	Reserved

Broadcast on CIDR network

IP address = **octet1.octet2.octet3.octet4/nw_bits**

MASK

11...1	00...0
nw_bits	32 – nw_bits

MASK'

00...0	11...1
nw_bits	32 – nw_bits

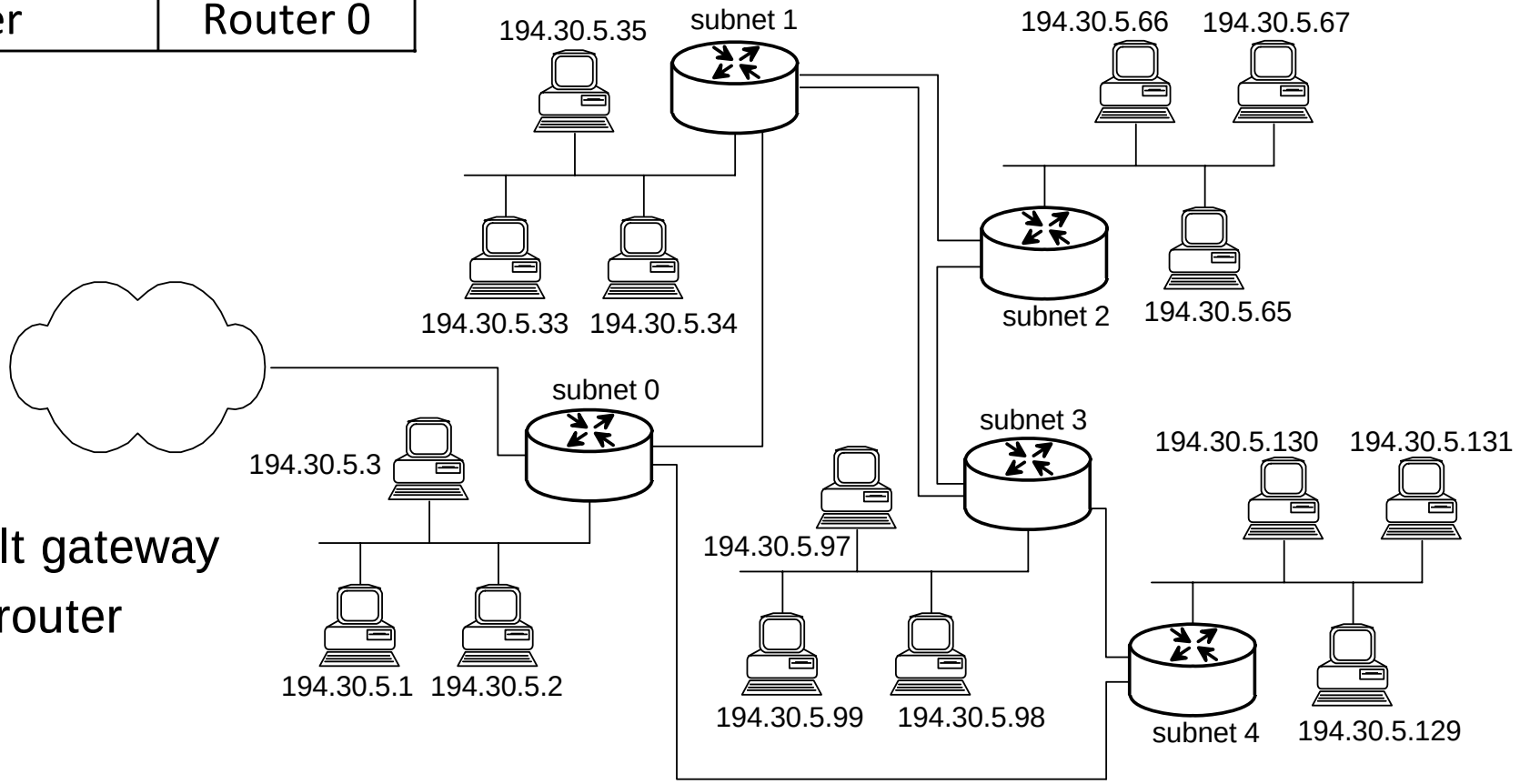
Network number = MASK AND IP

Broadcast = MASK' OR IP

CIDR Routing Table

Router 1 Table	
Destination	**Interface**
194.30.5.32/27	LAN
194.30.5.64/27	Router 2
194.30.5.96/27	Router 3
194.30.5.128/27	Router 3
Other	Router 0

Router 4 Table	
Destination	**Interface**
194.30.5.128/27	LAN
194.30.5.64/27	Router 3
194.30.5.96/27	Router 3
Other	Router 0



Router 0

Default gateway

Edge router

Routing Algorithms

Application of graph theory

- Model network as undirected graph G(N,E)
  - Nodes N = switches / routers
  - Edges E = one-hop links between nodes
  - Weight = metric (cost, distance) > 0
    - Weighted average of path parameters
    - Path length, queuing and transmission delay, cost, etc
- Shortest path routing by standard algorithms

Static routing

- Apply Dijkstra algorithm
- Link State Routing
- Requires knowledge of all weights

Dynamic routing

- Solve Bellman-Ford equation locally
- Nearest neighbor nodes exchange weight parameters
- Distance Vector Routing

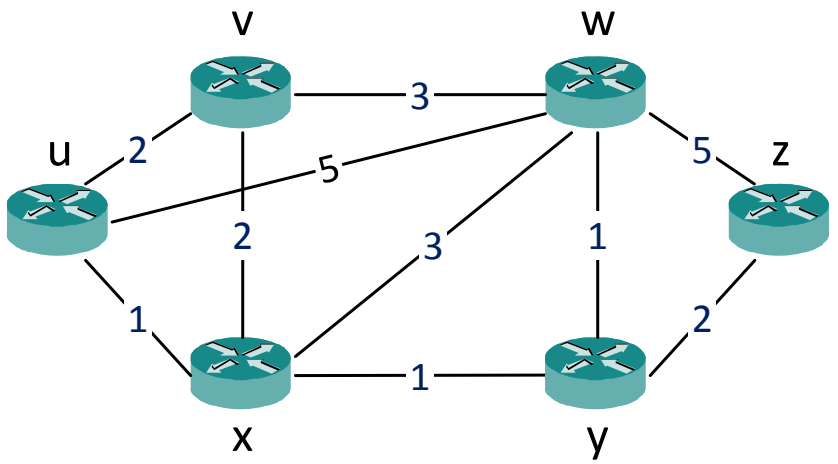
Router Graph

Nodes

N = {u, v, w, x, y, z}

Weights

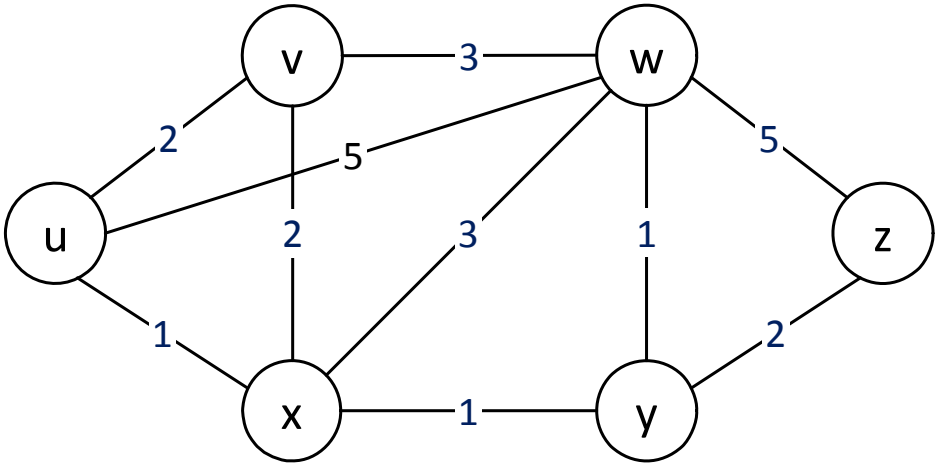
c(i,j) = link metric from i to j



Optimal path

Path from x to y with minimum cost

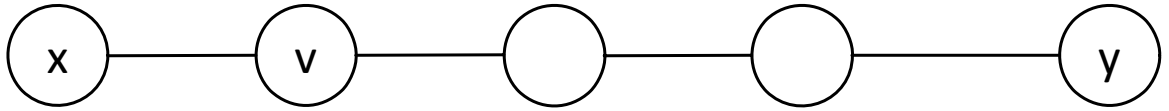
$d_x(y)$ = metric of optimal path from x to y



Bellman-Ford Equation

Principle of optimality

Node v on optimal path from x to y $\Rightarrow$ xv and vy = optimal paths



Corollary

$$xvy \text{ optimal} \Rightarrow d_x(y) = c(x,v) + d_v(y) \Rightarrow c(x,v) = d_x(v)$$

Bellman-Ford equation

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

Basis for static and dynamic routing algorithms

Dijkstra Algorithm

Initialize

$$N' = \{u\} \qquad D(v) = \begin{cases} c(u,v), & uv \text{ is a link} \\ \infty, & \text{otherwise} \end{cases} \qquad p(v) = \begin{cases} u, & uv \text{ is a link} \\ 0, & \text{otherwise} \end{cases}$$

While $(N \backslash N' \neq \phi)$

find $w \in N \backslash N'$ with minimum $D(w)$

move w from N to N'

for $(v \in N \backslash N')$ {

$L = D(w) + c(v,w)$

if $(L < D(v))$ {

$D(v) = L$

$p(v) = w$

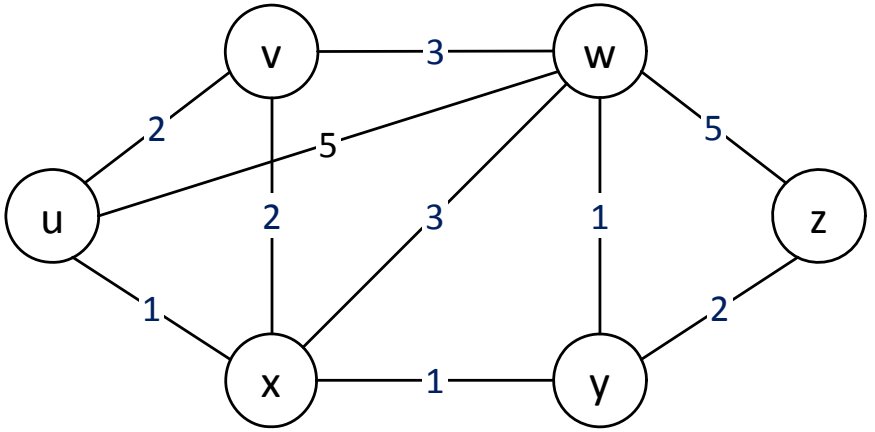
}

}

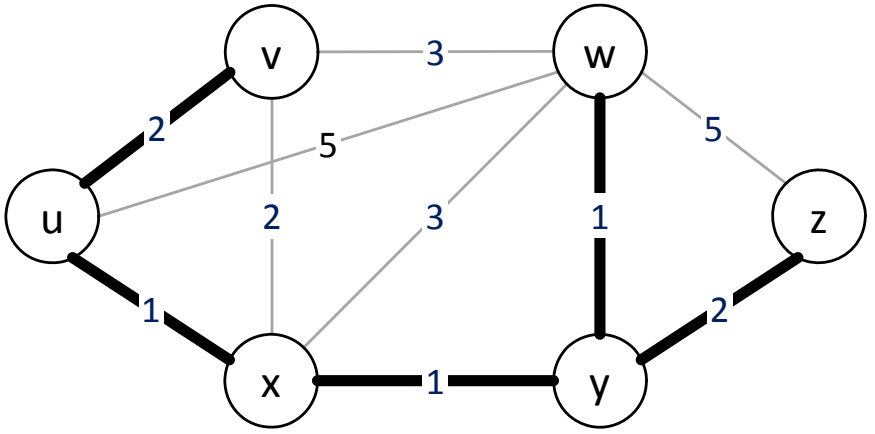
Dijsktra Example

Source u

N = {u, v, w, x, y, z}



N	N'	Minimum D	D{v, w, x, y, z}	p{v, w, x, y, z}
{v, w, x, y, z}	{u}		{2, 5, 1, ∞, ∞}	{u, u, u, 0, 0}
{v, w, y, z}	{u, **x**}	D(x) = 1	{2, 4, 1, 2, ∞}	{u, x, u, x, 0}
{w, y, z}	{u, **v**, x}	D(v) = 2	{2, 4, 1, 2, ∞}	{u, x, u, x, 0}
{w, z}	{u, v, x, **y**}	D(y) = 2	{2, 3, 1, 2, 4}	{u, y, u, x, y}
{z}	{**w**, u, v, x, y}	D(w) = 3	{2, 3, 1, 2, 4}	{u, y, u, x, y}



Distance Vector Routing

Routing table in each node

Index = known nodes

Entry = metric to node

Information exchange

Nodes pass routing tables to one-hop neighbors

Table entries

One-hop node

Metric = measured cost parameters

Multi-hop nodes

Metric = min{measured cost to neighbor + neighbor entry to node}

Bellman-Ford routing

Assume route information from node v to node y is $d_v(y)$

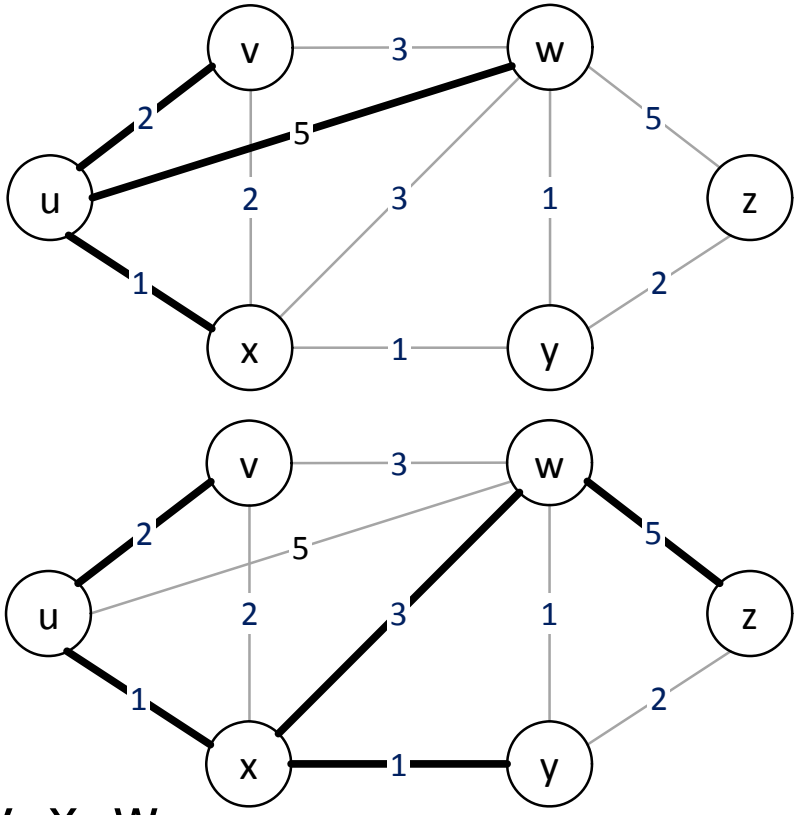
At node x choose route to y via v where

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

DV Example — Exchange 1

Initial tables

	source					
dest	u	v	w	x	y	z
u	—	2	5	1	—	—
v	2	—	3	2	—	—
w	5	3	—	3	1	5
x	1	2	3	—	1	—
y	—	—	1	1	—	2
z	—	—	5	—	2	—



Exchange 1

Node u gets routing information from v, x, w

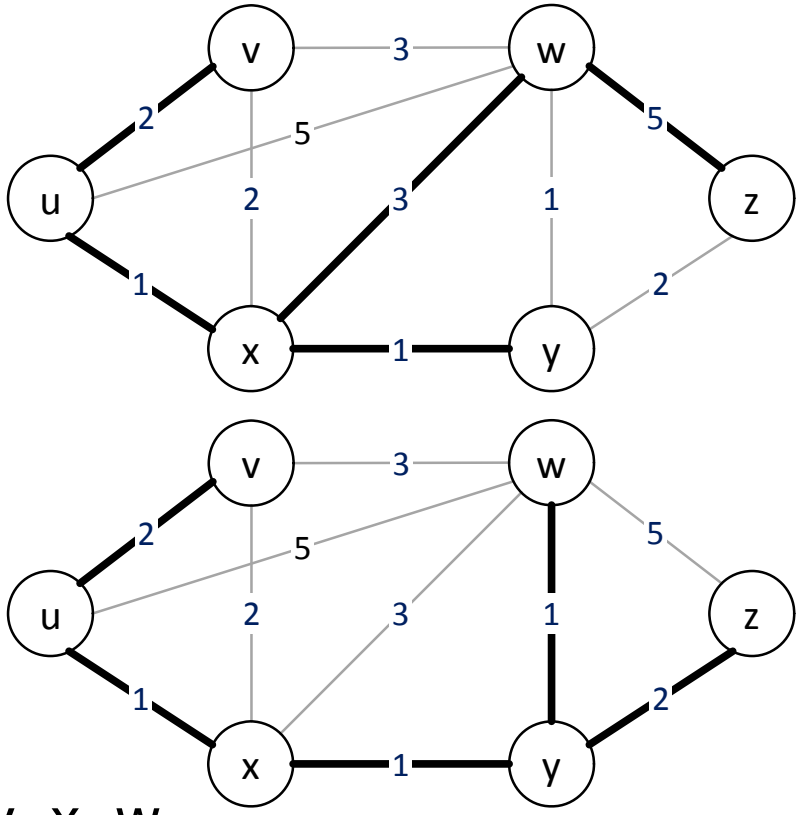
	Metric	Next hop
u	—	—
v	2	v
w	4	x
x	1	x
y	2	x
z	9	w

v	w	x
2	5	1
—	3	2
3	—	3
2	3	—
—	1	1
—	5	—

DV Example — Exchange 2

After Exchange 1 at v, x, w

	v	w	x
u	2,v	4,x	1,u
v	—	3,v	2,v
w	3,w	—	2,y
x	2,x	2,y	—
y	**3,x**	1,y	1
z	**8,w**	3,y	3,y



Exchange 2

Node u gets routing information from v, x, w

	Metric	Next hop
u	—	—
v	2	v
w	**3**	x
x	1	x
y	2	x
z	**4**	x

v	w	x
2	**4**	1
—	3	2
3	—	**2**
2	**2**	—
3	1	1
8	**3**	**3**

Spanning Tree

Spanning tree

Subgraph of G(N,E') $\subseteq$ G(N,E)

Includes every node

Unique path from root node u to any node

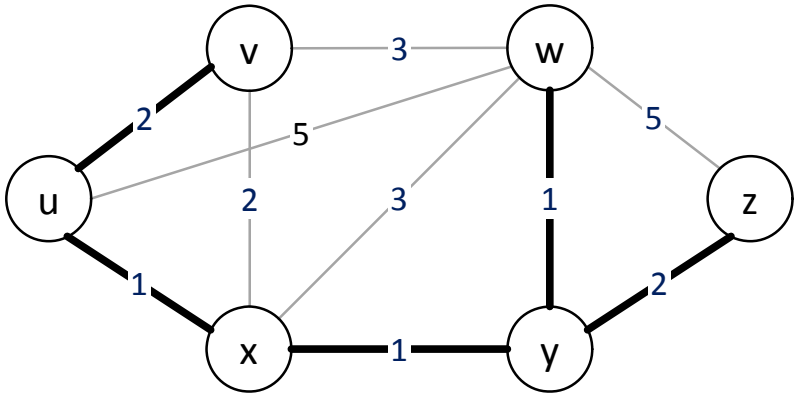
No cycles (loops)

Dijkstra and DV routing

Converge to spanning tree for simple graphs

Minimal spanning tree

Minimal cost from root to any node



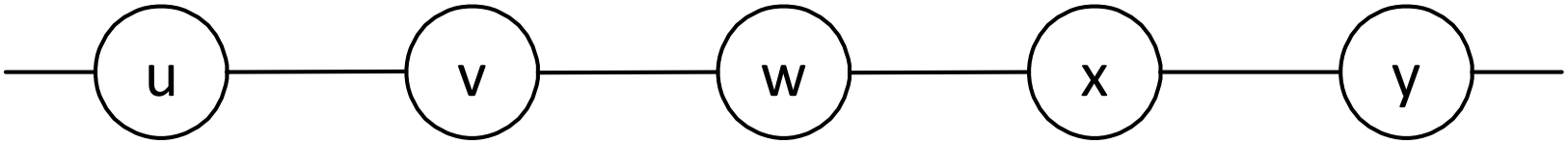
Distance Vector Metric Updates

Good news travels fast

Metric

Hops to node u

Link uv recovers from failure

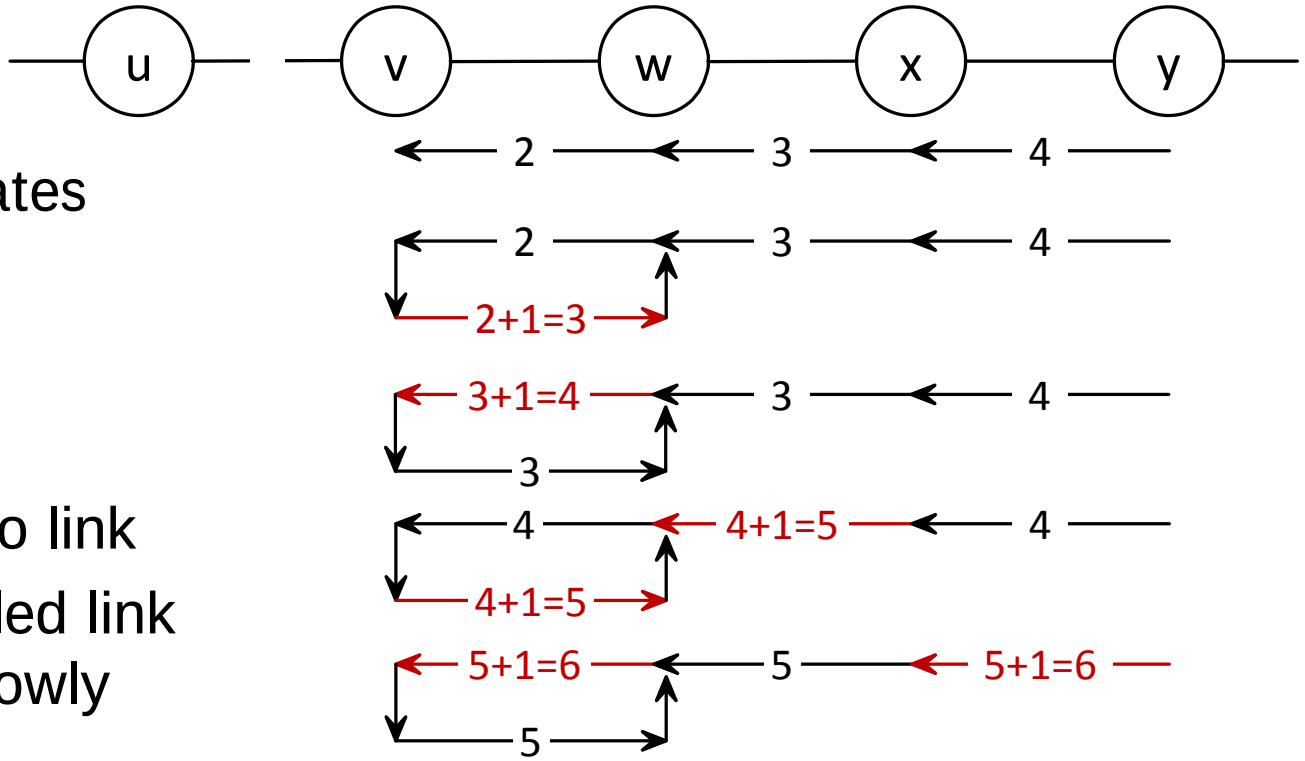


	Routing Table Entries for Node u			
exchange	v	w	x	y
0	**1,u**	∞	∞	∞
1	1,u	**2,v**	∞	∞
2	1,u	2,v	**3,w**	∞
3	1,u	2,v	3,w	**4,x**

Distance Vector Metric Updates

Bad news travels slowly

Link uv fails



Circular table updates

Count to infinity

N_{max} hops

Metric > N_{max} ⇒ no link

Information on failed link propagates slowly

	Routing Table Entries for Node u			
exchange	v	w	x	y
0	1,u	2,v	3,w	4,x
1	**3,w**	2,v	3,w	4,x
2	3,w	**4,v**	3,w	4,x
3	**5,w**	4,v	**5,w**	4,x
4	5,w	**6,v**	5,w	**6,x**

Internet Routing

Autonomous System (AS)

Nodes managed by one organization

Hierarchical routing

Interior Gateway Protocols (IGP)

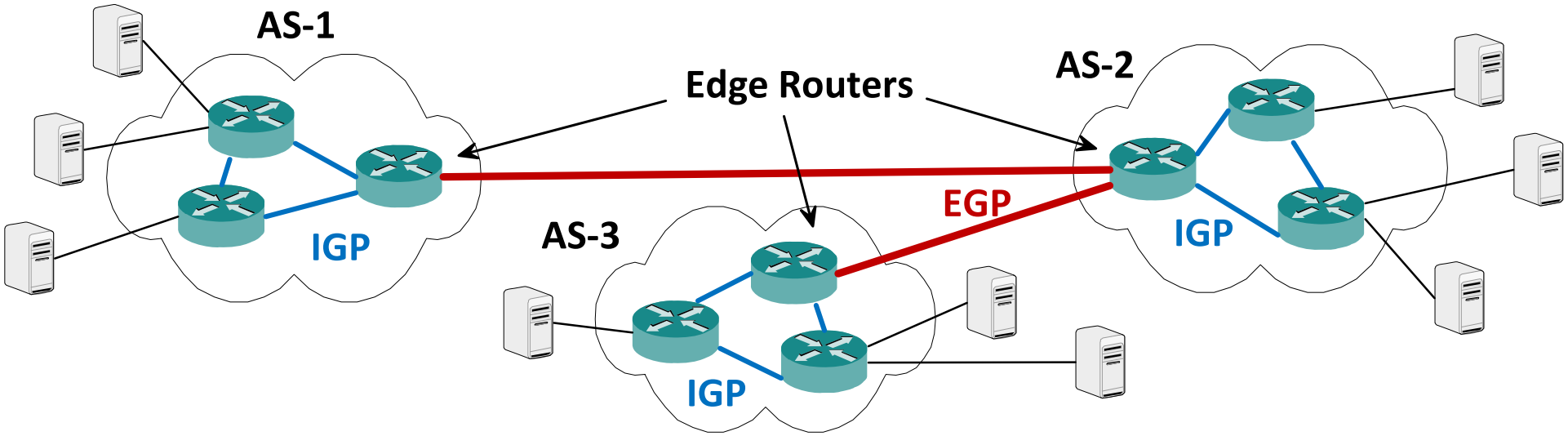
Routing protocols within one AS (Intra-AS)

Exterior Gateway Protocols (EGP)

Routing protocols between ASs (Inter-AS)

Edge router (gateway router)

Router within AS linked to router in different AS



Internal Gateway Protocols (IGP)

Intra-AS tasks

Optimize internal routing

Exchange routing tables with IGP nodes

Link to primary gateway (edge router)

IGP protocols

Routing Information Protocol (RIP)

RFC 1058, RFC 2453

Open Shortest Path First (OSPF)

RFC 2328

Interior Gateway Routing Protocol (IGRP)

Cisco proprietary

Routing Information Protocol (RIP)

Older protocol

Introduced in BSD-UNIX (1982)

Distance Vector (DV) algorithm

Routing information sent as data in UDP segments

DV advertisements

Routing information exchanged with 1-hop neighbor nodes

Exchanged every 30 seconds

No routing update in 180 seconds $\Rightarrow$ node = unreachable

Metric = hops

Table entry = 1 hop to neighbor + neighbor table entry

$N_{max} = 15$

$N > N_{max} \Rightarrow$ node = unreachable

Open Shortest Path First (OSPF)

Newer protocol (1998)

- Link State algorithm
- Independent network layer protocol (not data over UDP)

LS advertisements (LSAs)

- Start
  - Routing metrics exchanged among 1-hop neighbors
- Continue
  - Routing tables synchronized among all routers in AS

Shortest Path First (SPF)

- Dijkstra algorithm among routers in AS
- Sub-optimal next hop permitted for load balancing

Metrics

- Hops, speed, delays, connections from destination
- Multiple service-dependent metrics permitted

Hierarchical Open Shortest Path First (OSPF)

Divide AS into hierarchy of areas

Routers in area perform OSPF among themselves

Reduces overall routing information traffic

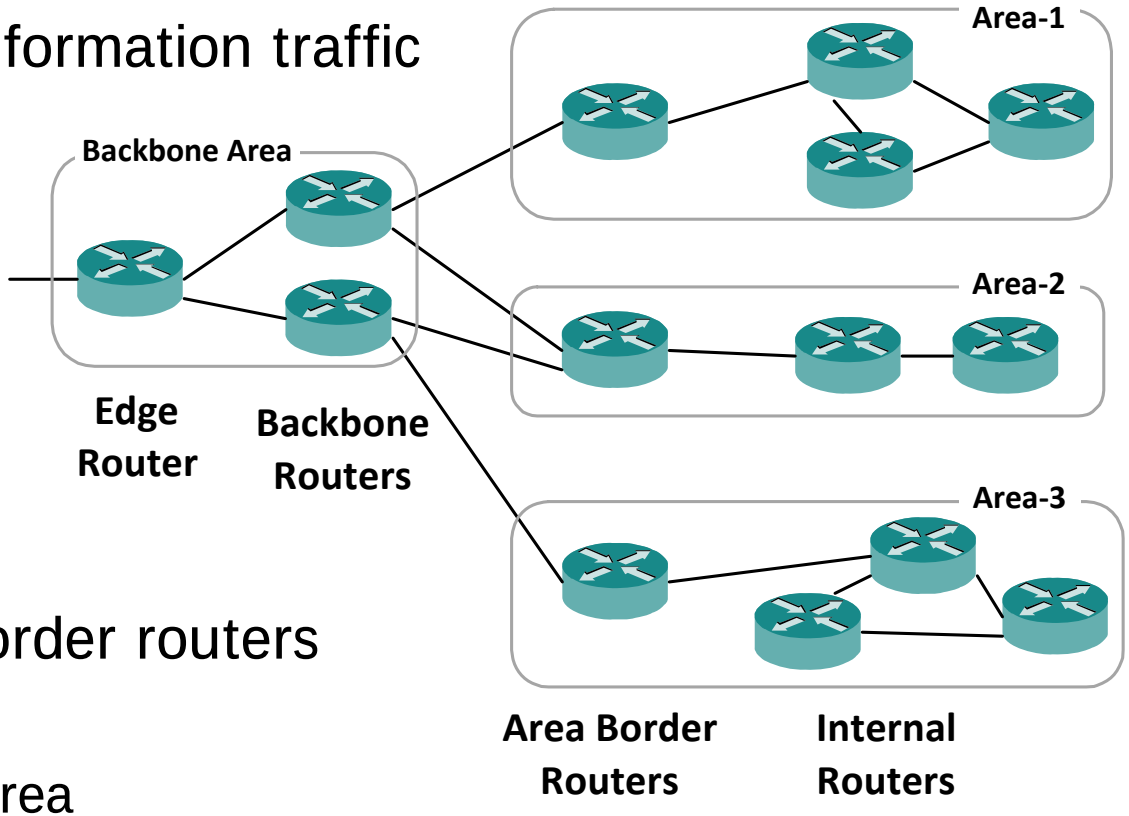
Area hierarchy

Edge router

Backbone

Area border routers

Internal area routers



LS advertisements (LSAs)

Internal routers + area border routers

- Perform full OSPF
- Optimal routing within area

Area border routers + backbone

- Exchange summary metrics
- Basic topology

Backbone + edge perform OSPF as an area

Exterior Gateway Protocols (EGP)

Inter-AS tasks

Exchange routing information among EGP nodes

Determine routes for inter-AS traffic

Manage economic, political, security considerations

Avoid global instabilities (routing cycles)

EGP protocols

Exterior Gateway Protocol (EGP)

- Obsolete
- RFC 827, RFC 904

Border Gateway Protocol (BGP)

- RFC 4271

Border Gateway Protocol (BGP)

Routing hierarchy

eBGP

Exchange routing information between AS edge routers

Determine routes for inter-AS traffic

iBGP

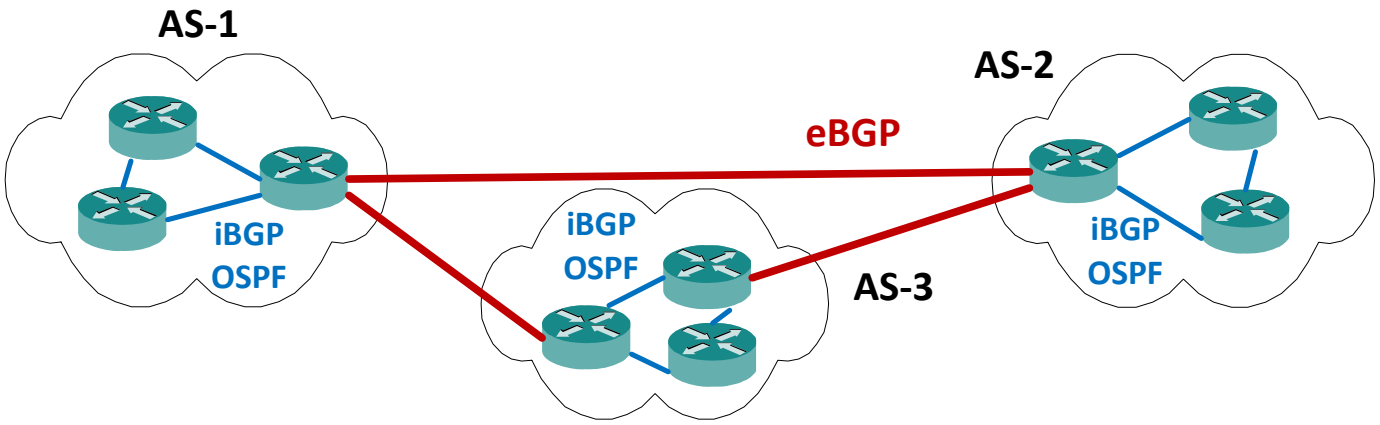
Exchange routing information within AS

Determine preferred gateway for inter-AS traffic

OSPF (or other IGP)

Exchange routing information within AS

Determine routes for intra-AS traffic

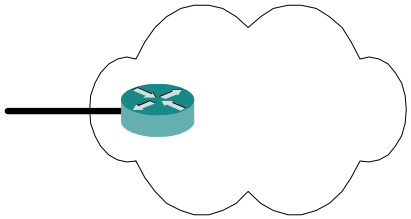


Border Gateway Protocol (BGP)

AS classification

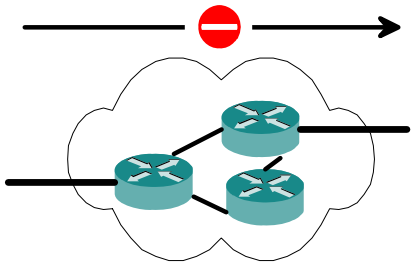
Stub network

- 1 edge router
- Cannot carry transit traffic
  - All incoming / outgoing traffic has dest / src in AS



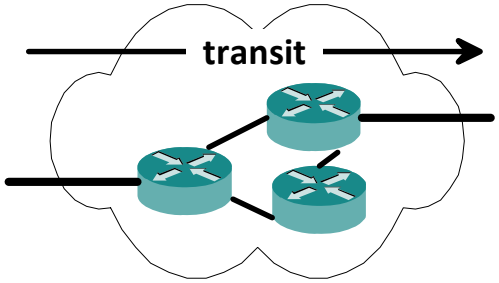
Multi-homed network

- More than 1 edge router
- Refuses transit traffic
  - All incoming / outgoing traffic has dest / src in AS
- Typically large corporate network or commercial ISP
  - Unwilling to accept traffic burden



Transit network

- More than 1 edge router
- Accepts transit traffic
- Internet backbone



AS reachable via

- One hop link
- Transit network

Border Gateway Protocol (BGP)

Routing information exchange

BGP speaker	Router supporting BGP
BGP peers	Two BGP speakers exchanging routing information
BGP session	Persistent TCP session between two BGP peers Used to exchange routing information
AS number (ASN)	16-bit number to identify AS
Network Layer Reachability Information (NLRI)	CIDR description of reachable Internet destination AS
AS route	NLRI with network attributes AS path List of ASNs describing path to NLRI Next-Hop IP address of next hop edge router

Border Gateway Protocol (BGP)

Path selection

No routing cycles

Router eliminates path that includes itself as transit network

Local preference attribute

Administrator policy sets preference value for paths

Lower preference paths eliminated

Shortest AS path

Path with least transit networks

Paths with longer AS path eliminated

Closest Next-Hop router

Possibly multiple routes to next AS

Determine minimum cost link to router in next AS

Routes with higher cost route to next hop router eliminated

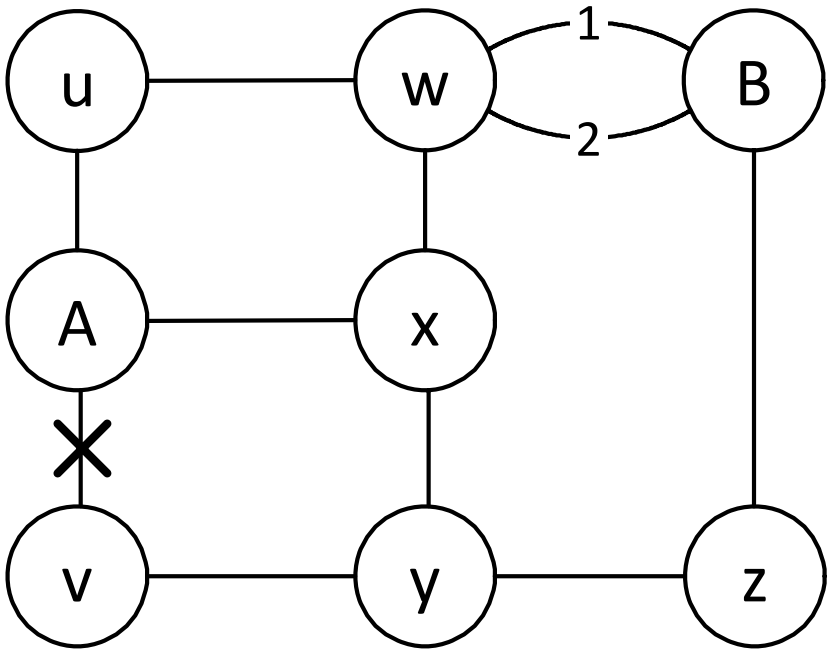
Border Gateway Protocol (BGP)

No routing cycles

Path from A to B

A receives paths from peers

Peer	Peer Recommends
u	uwB
x	xwB
v	vAuwB



No loops

A eliminates path AvAuwB

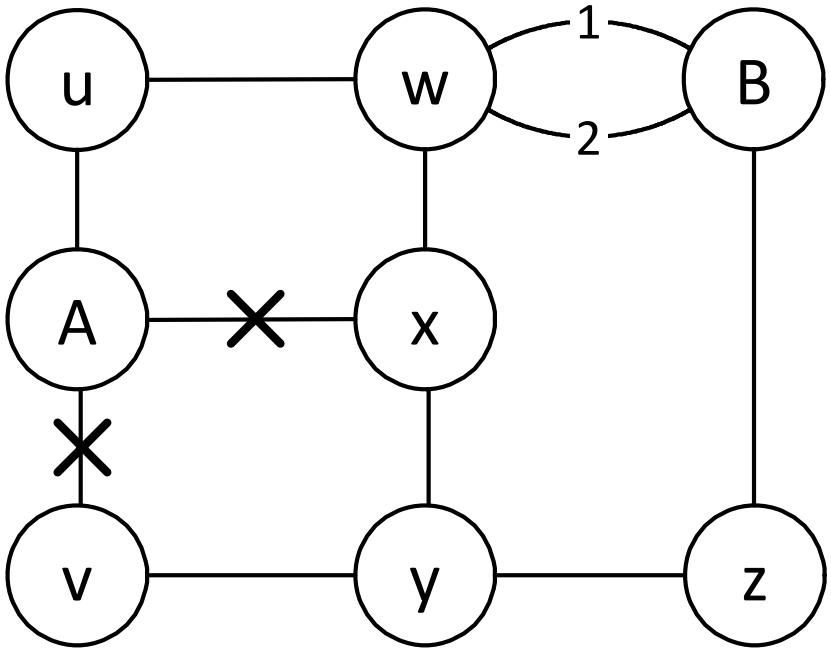
Border Gateway Protocol (BGP)

Local preference attribute

Path from A to B

A receives paths from peers

Peer	Peer Recommends	Local Preference
u	uwB	1
x	xwB	2
v	vAuwB	2



No loops

A eliminates path AvAuwB

Local administrator sets preferences

AS u preferred to AS x as transit network

Example — data security questions

A eliminates path AxwB

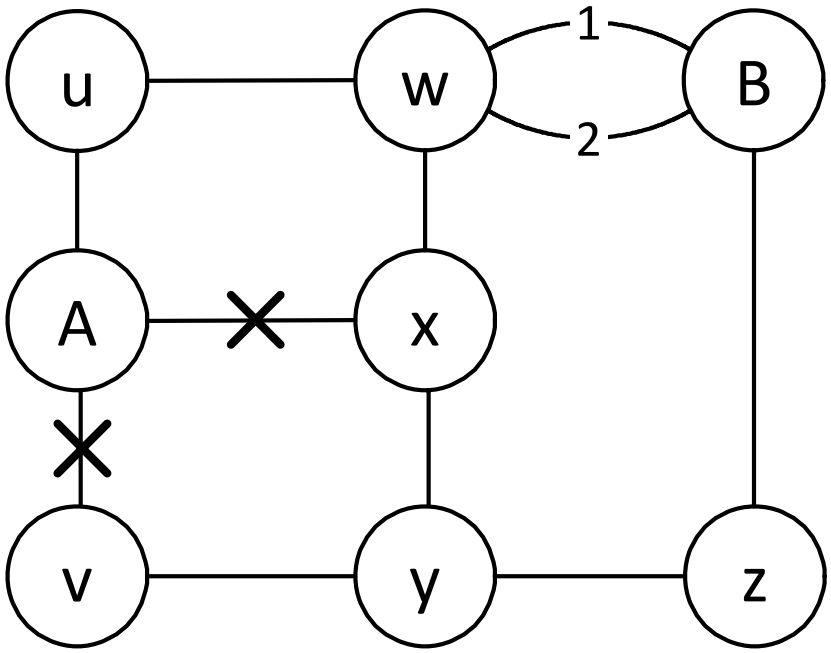
Border Gateway Protocol (BGP)

Shortest AS path

Path from A to B

A receives update from v

Peer	Peer Recommends	Local Preference
u	uwB	1
x	xwB	2
v	vyzB	2



Local administrator sets preferences

AS u preferred to AS x as transit network

Example — data security questions

A eliminates path AxwB

Shortest AS path

AvyzB longer than AuwB

A eliminates AvyzB

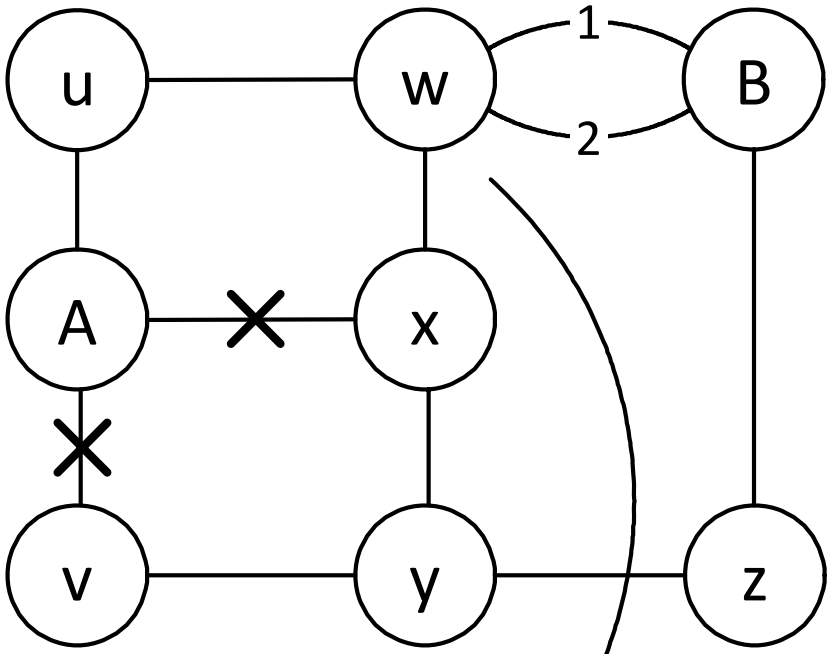
Border Gateway Protocol (BGP)

Closest Next-Hop router

Path from A to B

A receives update from u

Peer	Peer Recommends	Local Preference
u	uwB	1
x	xwB	2
v	vyzB	2



Local administrator sets preferences

AS u preferred to AS x as transit network

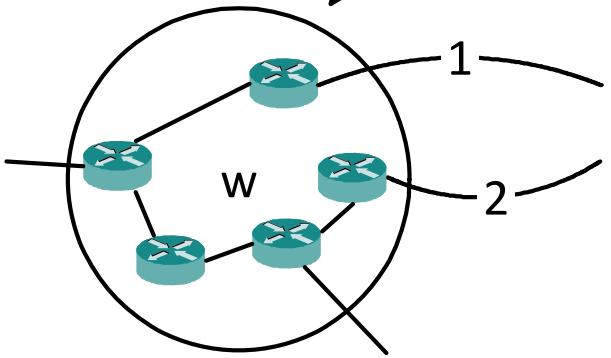
Example — data security questions

A eliminates path AxwB

Shortest AS path

AvyzB longer than AuwB

A eliminates AvyzB



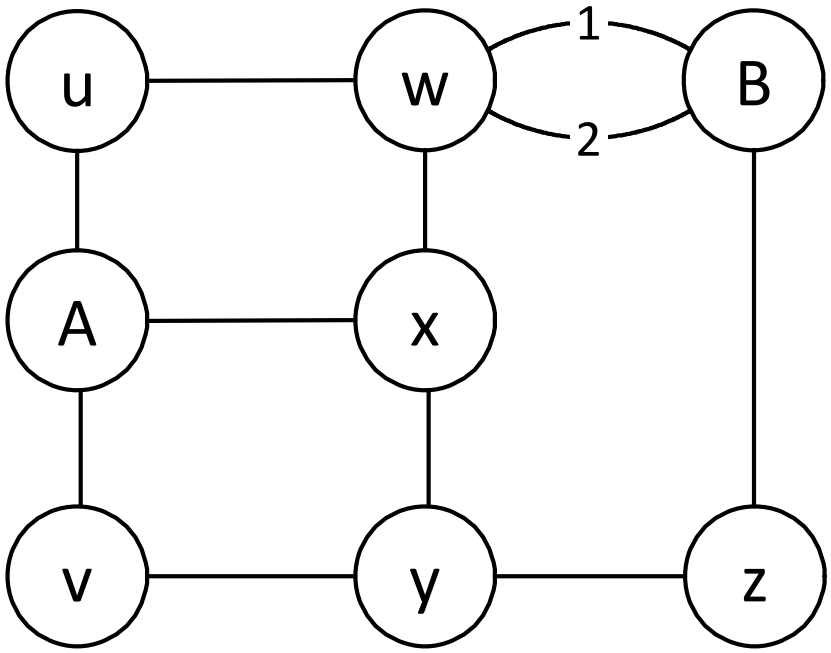
Path 1 closer than path 2

Border Gateway Protocol (BGP)

Refusing transit traffic

Transit policy

- Commercial ISP A
  - Revenue from attached customers
  - Customers require forwarding
- Transit traffic
  - Adds load to ISP network
  - Produces no revenue



A does not advertise

- Connection Ax to u or v
- Connection Av to u or x
- Connection Au to x or v

Broadcast / Multicast Services

Broadcast

- Forward datagrams to all nodes

Multicast

- Forward datagrams to group of nodes

Local broadcast / multicast

- Among nodes attached to one router
- Server advertisements
  - DHCP
  - DNS
  - Router services
- Handled as layer 3 / layer 2 transactions

Global broadcast / multicast

- Among multiple routers
- File sharing
- Content distribution
- Media streaming

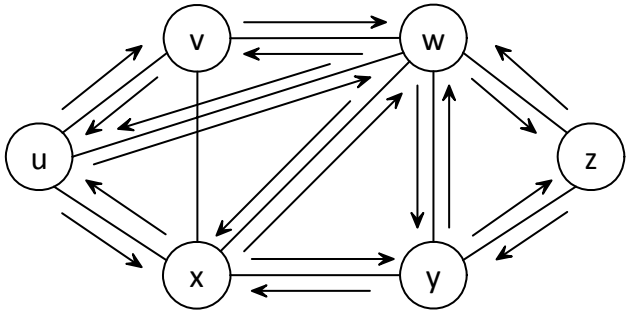
Broadcast Issues

Node-level broadcast

- Multiple unicast
- Requires list of all nodes
- Not practical

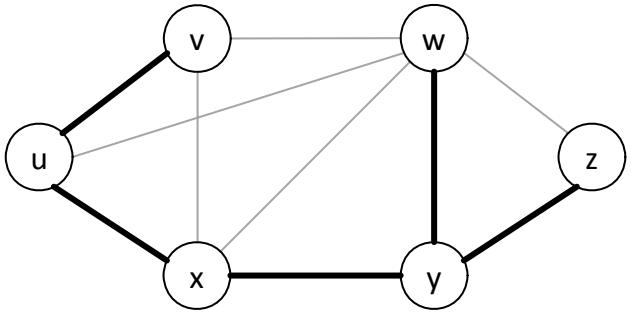
Uncontrolled flooding

- Each router forwards all packets to all routers
- Broadcast storm
  - Traffic multiplies exponentially
  - Network paralyzed



Controlled flooding methods

- Tag packets
  - Transmit each packet one time
- Time to Live (TTL)
  - Delete packets after N_max hops
- Broadcast only on spanning tree
  - No cycles in graph



Spanning Tree Broadcast

Reverse Path Forwarding (RPF)

Only forward packet arriving on default node-to-source path

Node uses default routing table

Does not know full spanning tree

Example

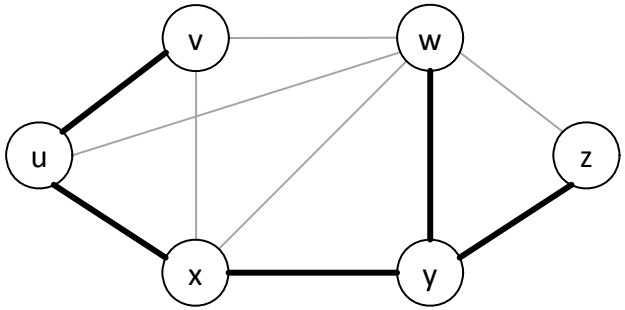
Node broadcast u arrives at node x

Arrives on u-x

Node x forwards to y on x-y

Arrives on y-x, w-x, v-x

Node x deletes



Node x Routing Table	
u	u
v	
w	y
y	
z	

Center-based forwarding

Designated center node

Nodes send join messages to center node by conventional routing

Center node broadcasts to group members

Multicast

Multicast group

Subset of all routers / hosts receiving multicast datagrams

Multicast router

Maintains group membership information for multicast distribution

Multicast address

Class D address refers to multicast group

Multicast datagrams addressed to multicast address

Multicast routers distribute to group members

Internet Group Management Protocol (RFC 3376)

Nodes report multicast group memberships to multicast routers

Protocol Independent Multicast (PIM)

Dense mode — multicast routers closely distributed

- RPF spanning tree construction

Sparse mode — multicast routers widely distributed

- Center-based forwarding

Network Address Translation (NAT)

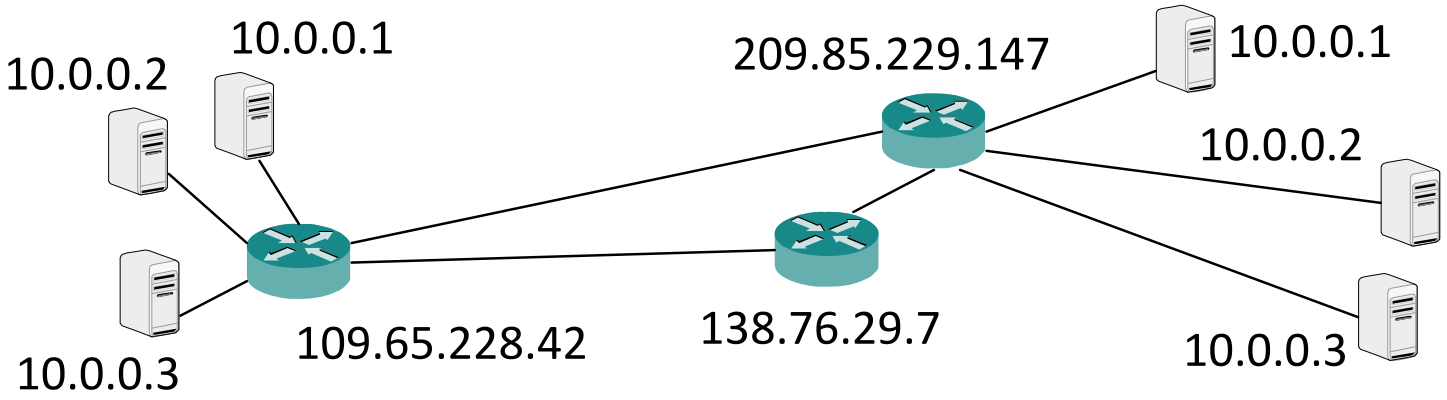
Router

- Receives IP datagram
- Exchanges IP source / destination address in header
- Forwards datagram
- Maintains list of translations

Autonomous systems

- Allocate private network addresses internally
  - 10.0.0.0 and 192.168.0.0
- Only gateway and backbone devices require unique IP addresses
  - Local nodes on AS re-use private addresses

Example



Network Address Translation (NAT)

Translation details

No translation required on traffic internal to AS

- Local addresses = real IP addresses

Local node to external node

- Outgoing packet
  - Source endpoint = local IP address + application source port number
  - Dest endpoint = remote IP address + well-known port number
- Gateway router NAT
  - Replaces
    - Local IP address with gateway IP address
    - Local application source port with unique unused NAT port
  - Records mapping
    - NAT source port → local IP address + source port number
  - Forwards packet

External node to local node

- Gateway router replaces NAT port → local IP + app source port

Network Address Translation (NAT)

Example

Host A HTTP request to Server S

Source port = **1025** Source IP = **10.0.0.2**

Destination port = 80 Destination IP = 209.85.229.147

Outgoing NAT at Gateway router B

Source port = **3745** Source IP = **109.65.228.42**

Destination port = 80 Destination IP = 209.85.229.147

Server S HTTP response to Host A

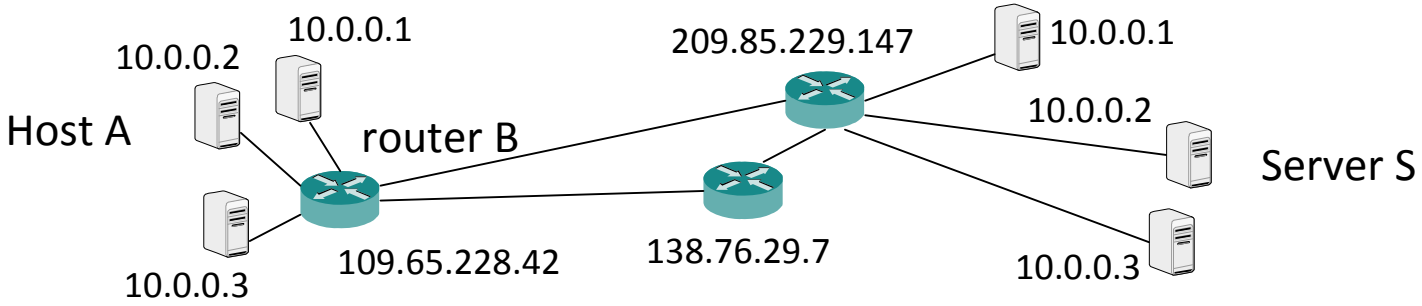
Source port = 80 Source IP = 209.85.229.147

Destination port = **3745** Destination IP = **109.65.228.42**

Incoming NAT at Gateway router B

Source port = 80 Source IP = 209.85.229.147

Destination port = **1025** Destination IP = **10.0.0.2**



Network Address Translation (NAT)

Traversal problem

Problem

How to operate service behind NAT

Solutions

Static router referral

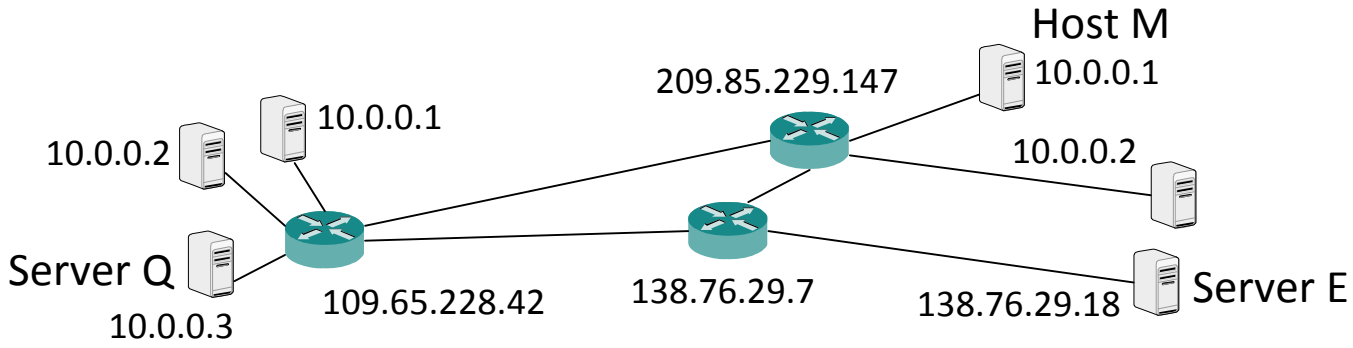
- Define service port on Server Q
- Refer all incoming traffic at gateway for service port to Server Q

Relay server

- Server Q connects to Server E with public address (no NAT)
- Host M requests service from Server Q via Server E
- Server Q replies via Server E

Universal Plug and Play (UPnP)

- Specialized protocols for NAT traversal



Assistive Protocols used with IP Layer

Domain Name System (DNS)

Convert between node name and network address

Address Resolution Protocol (ARP)

Convert between network and hardware addresses

Discover local subnet topology

Dynamic Host Configuration Protocol (DHCP)

IP address allocation on request from DHCP server

- Server typically in gateway router

Server allocates IP address from pool of available addresses

DHCP message types

- DHCP server discovery (host broadcast)
- DHCP offer (server response)
- DHCP request (host request)
- DHCP ACK (server provides address)

IP Configuration Example

```
c:\>ipconfig /all
Windows IP Configuration
        Host Name . . . . . . . . . . . . : ******
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No
        DNS Suffix Search List. . . . . . : ******
Ethernet adapter Local Area Connection:
        Connection-specific DNS Suffix  . : ******
        Description . . . . . . . . . . . : ******
        Physical Address. . . . . . . . . : ******
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 10.0.0.5
        Subnet Mask . . . . . . . . . . . : 255.0.0.0
        Default Gateway . . . . . . . . . : 10.0.0.138
        DNS Servers . . . . . . . . . . . : 10.0.0.11
                                            10.0.0.138
c:\>arp -a
Interface: 10.0.0.5 --- 0x2
  Internet Address      Physical Address      Type
  10.0.0.7              70-1a-04-95-a4-4f     dynamic
  10.0.0.11             00-50-fc-b1-6f-ff     dynamic
  10.0.0.138            5c-d9-98-06-26-36     dynamic
```

Internet Control Message Protocol (ICMP)

Control protocol

Network management information

Error reporting

Unreachable host / network / port / protocol

Echo request / reply

Packets carried in IP datagrams

Packet structure

Offset	0 – 7	8 – 15	16 – 23	24 – 31
0	Type	Code (subtype)	Checksum	
32	Rest of Header			
64	Padding			

Rest of Header — message specific

Padding — data field

Selected Message Types		
Type	Code	description
0	0	echo reply
3	0	dest network unreachable
	1	dest host unreachable
	2	dest protocol unreachable
	3	dest port unreachable
	6	dest network unknown
	7	dest host unknown
8	0	echo request
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Ping

Client application program

Client sends echo request packets to destination IP address

Type = 8 Code = 0 16-bit ID number 16-bit sequence number

OS dependent

Linux ping

Default = continuous packets with 56 padding bytes

Windows ping.exe

Default = 5 packets with 32 padding bytes

Ping server (OS service)

Responds to each echo request packet with echo rely packet

Type = 0 Code = 0 16-bit ID number 16-bit sequence number

Ping client

Measures round trip time (RTT)

Reports packet losses, RTTs and average RTT

UDP ping

Uses UDP instead of ICMP

Ping Example

```
c:\>ping www.hadassah.ac.il

Pinging hathi.hadassah.ac.il [212.179.79.228] with 32 bytes of data:

Reply from 212.179.79.228: bytes=32 time=32ms TTL=119
Reply from 212.179.79.228: bytes=32 time=70ms TTL=119
Reply from 212.179.79.228: bytes=32 time=135ms TTL=119
Reply from 212.179.79.228: bytes=32 time=83ms TTL=119

Ping statistics for 212.179.79.228:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 32ms, Maximum = 135ms, Average = 80ms
```

Traceroute

Client application program

- Multiple echo request packets to destination IP address
  - First packet TTL = 1
  - Each subsequent packet has TTL++
- OS dependent
  - Linux traceroute / tracepath
    - UDP to unlikely port number (port typically not defined)
  - Windows tracert.exe
    - ICMP echo request packets

Intermediate routers

`if (--TTL == 0){delete ; ICMP type 11 to client}`

- ICMP packet carries router name + IP address
- ICMP sequence = list of intermediate routers (hops = 1, 2, ...)

Destination node

- UDP segment — ICMP port undefined (type 3 code 3) message
- ICMP packet — ICMP echo reply message

Traceroute Example

```
c:\>tracert www.hadassah.ac.il

Tracing route to hathi.hadassah.ac.il [212.179.79.228]
over a maximum of 30 hops:

  1     2 ms     1 ms     2 ms  10.0.0.138
  2    18 ms    19 ms    17 ms  bzq-179-37-1.static.bezeqint.net [212.179.37.1]
  3    18 ms    17 ms    17 ms  bzq-179-48-201.static.bezeqint.net [212.179.48.201]
  4    17 ms    17 ms    17 ms  bzq-179-80-241.static.bezeqint.net [212.179.80.241]
  5    17 ms    18 ms    17 ms  bzq-179-124-201.static.bezeqint.net [212.179.124.201]
  6    17 ms    17 ms    17 ms  bzq-179-124-138.static.bezeqint.net [212.179.124.138]
  7    19 ms    18 ms    18 ms  bzq-179-59-1.static.bezeqint.net [212.179.59.1]
  8    32 ms    36 ms    41 ms  10.20.110.18
  9   105 ms    84 ms    71 ms  bzq-179-79-228.static.bezeqint.net [212.179.79.228]
 10    73 ms    88 ms   150 ms  bzq-179-79-228.static.bezeqint.net [212.179.79.228]

Trace complete.

c:\>
```

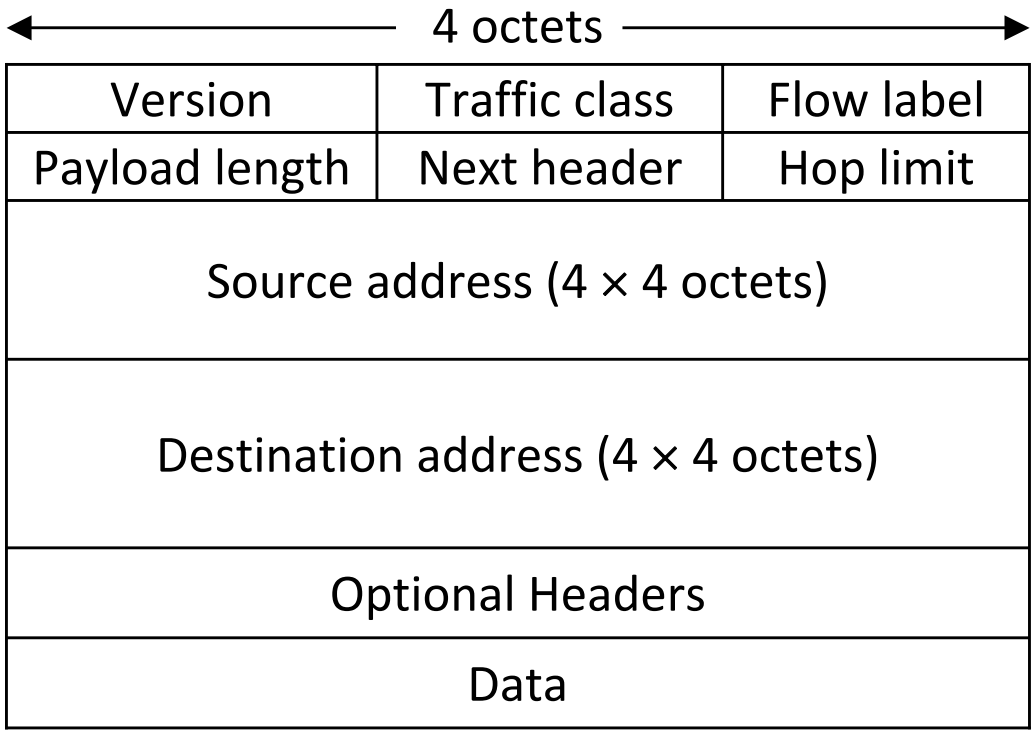
IPv6 (RFC 2460)

Motivation

- IPv6 address = 128 bits (16 octets)
- Service-specific fields
  - Quality of Service
  - Multicasting
  - Security
  - Mobility

Header

- Fixed 40-byte header
  - 10 × 4 bytes
- Optional header fields
  - Service-specific
- Next Header field
  - Identifies optional header or protocol of carried data
- Header daisy chain
  - Each optional IPv6 header contains Next Header field



IPv4 To IPv6 Transition

IPv4 routers

Replaced gradually

Work with IPv6 routers

IP tunneling

Encapsulate IPv6 datagram as data in IPv4 datagram

